ViennaNGS: A toolbox for building efficient next-generation sequencing analysis pipelines

Michael T. Wolfinger\textsuperscript{1,2,3}, Jörg Fallmann\textsuperscript{1}, Florian Eggenhofer\textsuperscript{1}, and Fabian Amman\textsuperscript{1,4}

\textsuperscript{1}Institute for Theoretical Chemistry, University of Vienna, Währingerstraße 17, 1090 Vienna, Austria
\textsuperscript{2}Center for Integrative Bioinformatics Vienna, Max F. Perutz Laboratories, University of Vienna, Medical University of Vienna, Dr. Bohr-Gasse 9, A-1030 Vienna, Austria
\textsuperscript{3}Department of Biochemistry and Molecular Cell Biology, Max F. Perutz Laboratories, University of Vienna, Dr. Bohr-Gasse 9, 1030 Vienna, Austria
\textsuperscript{4}Bioinformatics Group, Department of Computer Science and the Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstraße 16-18, 04107 Leipzig, Germany

December 19, 2014

Abstract

\textbf{Motivation:} Recent achievements in next-generation sequencing (NGS) technologies lead to a high demand for reusable software components to easily compile customized analysis workflows for big genomics data.

\textbf{Results:} We present ViennaNGS, an integrated collection of Perl modules focused on building efficient pipelines for NGS data processing. It comes with functionality for extracting and converting features from common NGS file formats, computation and evaluation of read mapping statistics, quantification and normalization of read count data, identification and characterization of splice junctions from RNA-seq data, parsing and condensing sequence motif data, automated construction of Assembly and Track Hubs for the UCSC genome browser and wrapper routines for a set of commonly used NGS command line tools.

\textbf{Availability:} The ViennaNGS Perl distribution is available through CPAN and GitHub at https://github.com/mtw/Bio-ViennaNGS

\textbf{Contact:} michael.wolfinger@univie.ac.at

\textbf{Supplementary Information:} Supplementary data for the ViennaNGS tutorial is available from http://rna.tbi.univie.ac.at/ViennaNGS
1 Introduction

Next-generation sequencing (NGS) technologies have influenced both our understanding of genomic landscapes as well as our attitude towards handling big biological data. Moreover, emerging functional genomics methods based on high-throughput sequencing allow investigation of highly specialized and complex scientific questions, which continuously poses challenges in the design of analysis strategies.

A set of NGS analysis pipelines are available for general (e.g. Förstner et al. (2014)), and specialized assays such as de novo motif discovery (Heinz et al. 2010). While these tools mostly cover the elementary steps of an analysis workflow, they often represent custom-tailored solutions that lack flexibility. Web-based approaches like Galaxy (Goecks et al. 2010) cover a wide portfolio of available applications, however they do not offer enough room for power-users who are used to the benefits of the command line.

The recently published HTSeq framework (Anders et al. 2014) and the biotoolbox suite provide library modules for processing high-throughput data. While both packages implement NGS analysis functionality coherently, we encountered use cases that were not covered by these tools.

2 Motivation

The motivation for this contribution emerged in the course of the research consortium "RNA regulation of the transcriptome" (Austrian Science Fund project F43), which is comprised of more than a dozen of experimental groups with various thematic backgrounds. In the line of this project it turned out that complex tasks in NGS analysis could easily be automated, whereas linking individual steps was very labour-intensive. As such, it became apparent that there is a strong need for modular and reusable software components that can efficiently be assembled into different full-fledged NGS analysis pipelines.

We present ViennaNGS, a Perl distribution that integrates high-level routines and wrapper functions for common NGS processing tasks. ViennaNGS is not an established pipeline per se, it rather provides tools and functionality for the development of NGS pipelines. It comes with a set of utility scripts that serve as reference implementation for most library functions and can readily be applied for specific tasks or integrated as-is into custom pipelines.

We share this publicly funded software package with the scientific community and provide extensive documentation, including a dedicated tutorial that demonstrates the core features and discusses some common application scenarios.

3 Description

The major design consideration for the ViennaNGS toolbox was to make available modular and reusable code for NGS processing in a popular scripting language. We therefore implemented thematically related functionality in separate Perl

\[^1\]https://code.google.com/p/biotoolbox
Figure 1: Schematic functionality of ViennaNGS modules. Core modules can be combined in a flexible manner to address different experimental setups and analysis objectives.

modules under the Bio namespace (Figure 1), some of which are based on BioPerl [Stajich et al., 2002] and the Moose object framework.

One of the most common tasks in NGS analysis is post-processing of BAM files, e.g. extraction of reads that align uniquely to a certain strand. ViennaNGS::Bam comes with routines for tasks like this. BAM files can be filtered for unique and multiple alignments, split by strand and the results can optionally be converted to bedGraph or bigWig formats for visualization purposes. Furthermore, ViennaNGS::BamStat collects high-level mapping statistics and produces publication-ready graphics.

ViennaNGS::SpliceJunc provides code for identification and characterization of splice junctions from short read mappers. It can detect novel splice junctions in RNA-seq data and generate visualization files. While we have focused on processing the output of segemehl [Hoffmann et al., 2014], the module can easily be extended for other splice-aware split read mappers.

Another major component of NGS post-processing is proper visual representation of mapped sequencing data. ViennaNGS::UCSC addresses this issue, aiming at two common visualization tasks: Deployment of custom organism databases in local mirrors of the UCSC Genome Browser and automated generation of UCSC Assembly and Track Hubs [Raney et al., 2014] from genomic sequence and annotation files.

ViennaNGS::AnnoC is a lightweight annotation converter for non-spliced ge-
onomic intervals whereas ViennaNGS::MinimalFeature, ViennaNGS::Feature and ViennaNGS::FeatureChain are generic Moose-based classes for efficient manipulation of genomic features.

ViennaNGS::Util implements wrapper routines for third-party utilities like BEDtools (Quinlan and Hall, 2010), BigWig and BigBed tools (Kent et al., 2010) and a set of utility functions.

Finally, the ViennaNGS::Tutorial module illustrates the core routines with as set of common use cases.

4 Conclusion

We have successfully applied ViennaNGS components in the context of different genomics assays (Antic et al., 2014; Sedlyarov et al., 2014) in combination with the short read aligner segemehl (Hoffmann et al., 2009, 2014). It has also been used with Tophat (Trapnell et al., 2009) output very recently in a large scale transcriptome study of Ebola and Marburg virus infection in human and bat cells (unpublished data). Moreover, ViennaNGS will be used for automated UCSC genome browser integration in an upcoming version of TSSAR (Amman et al., 2014), a recently published approach for characterization of transcription start sites from dRNA-seq data.

Funding This work was funded by the FWF projects F43 and W1207-B09, the Research Platform "Decoding mRNA decay in inflammation" (University of Vienna) and BMBF project 0316165C.

References


